Docker

Datalogforeningen, 7. June 2016

Martin Mosegaard Amdisen

Praqma

Continuous Delivery & DevOps experts and evangelists

Tools & Automation experts. We help customers with practical implementation of their development process.
We don't chop wood - we sharpen axes!

7 years, 25 employees, offices in Copenhagen, Aarhus, Oslo & Stockholm

Events: Jenkins CI User Events, Continuous Delivery & DevOps Conferences, DayOfContainers, Automation Nights, Code Academy (code-conf.com)

Service partner to:

docker GitHub GO WITH THE FLOW Jenkins Atlassian

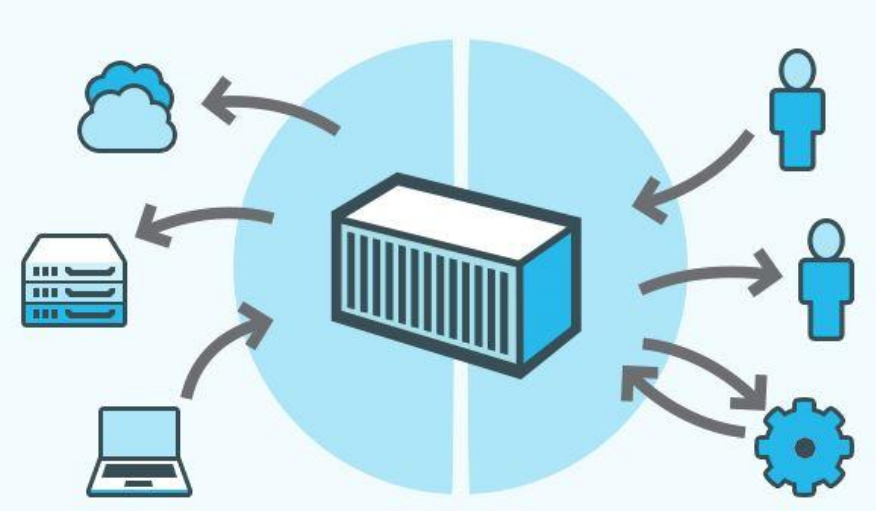
Agenda

- Introduction to Docker and containers
- Using Docker
- Docker and DevOps
- State of adoption
- Demo

PRAQMA

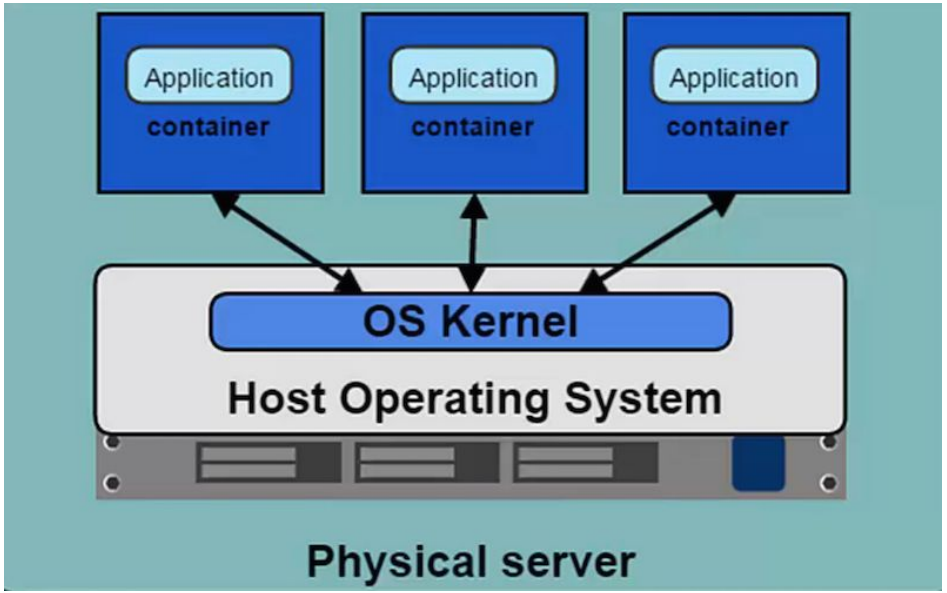
What is Docker?

- A platform to “build, ship, and run any app, anywhere” using container technology
- Many products and tools
  - Docker Engine
  - Docker Hub
  - Docker Machine
  - Docker Swarm
  - Docker Compose
  - ...

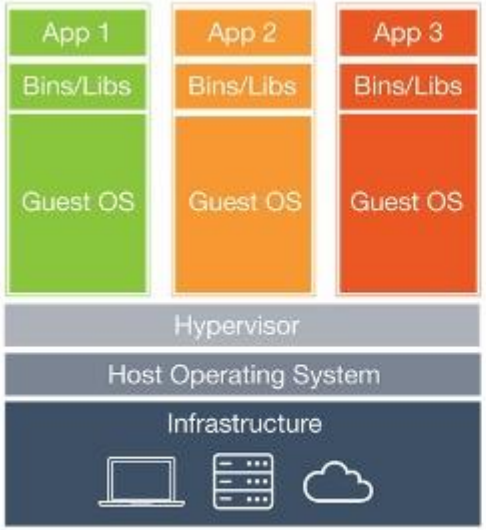


Introducing containers

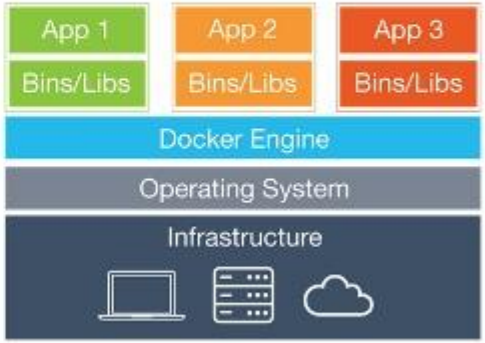
- Use the kernel on the host operating system to run multiple root file systems
- Each root file system is called a ***container***
- Each container also has its own
  - Processes
  - Memory
  - Devices
  - Network stack



Containers versus virtual machines



Virtual Machines

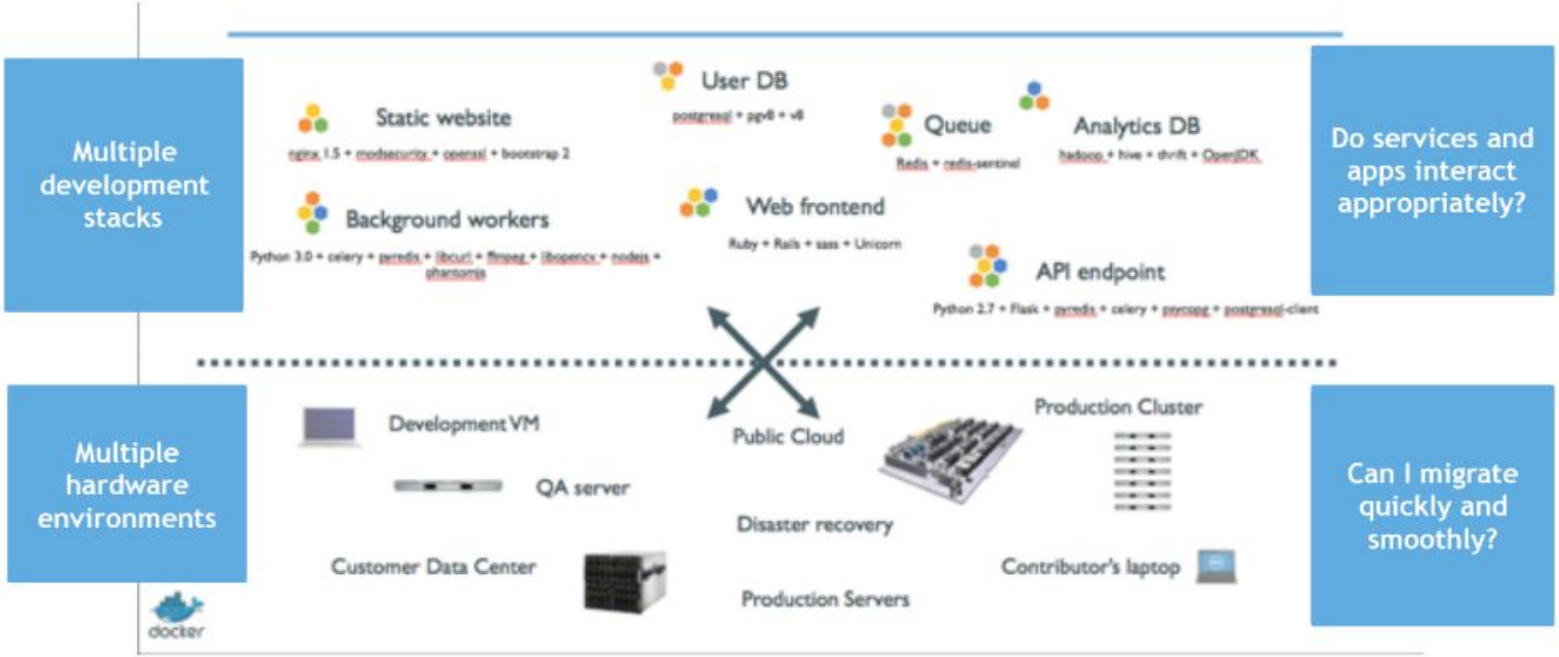


Containers

Why use Docker?

- Applications are no longer monolithic
- Service oriented architecture means different application stacks
- Services are decoupled and scaled out
- Deployment can become complex

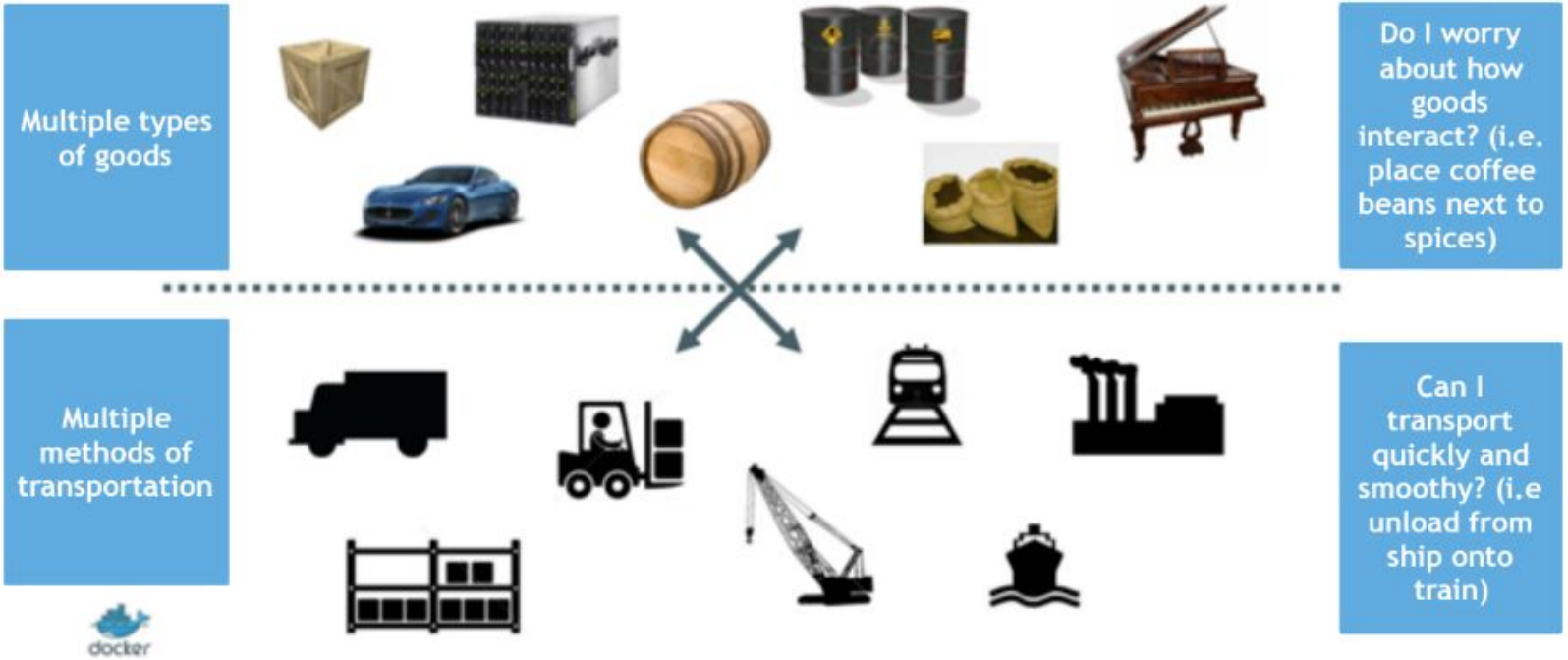
The deployment nightmare



The matrix from hell

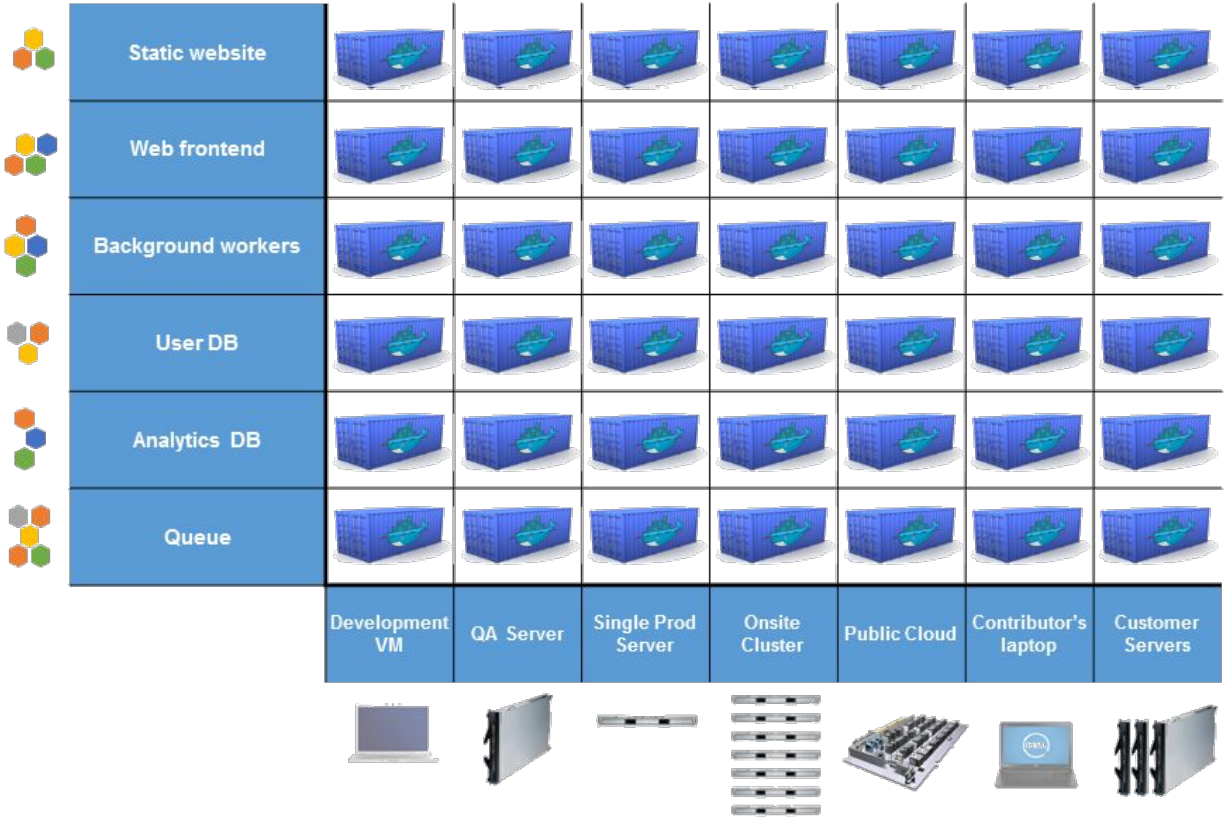
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

A shipping analogy



The shipping container

Solving the deployment matrix



PRAQMA

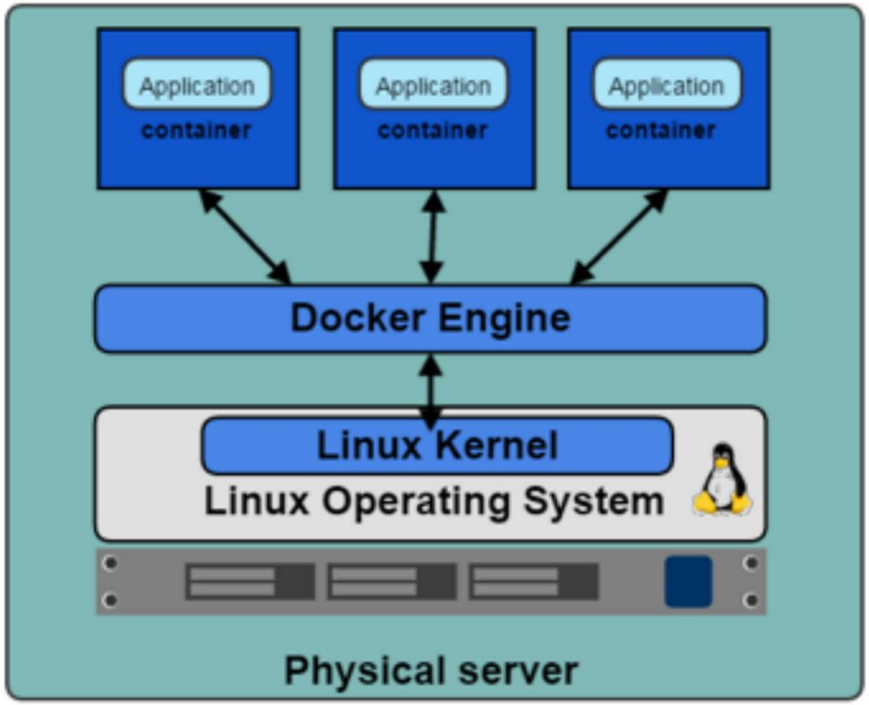
Benefits of Docker

- Separation of concerns
  - Developers focus on building apps
  - System admins focus on deployment
- Fast development cycle
- Application portability
- Scalability
- Infrastructure as code

PRAQMA

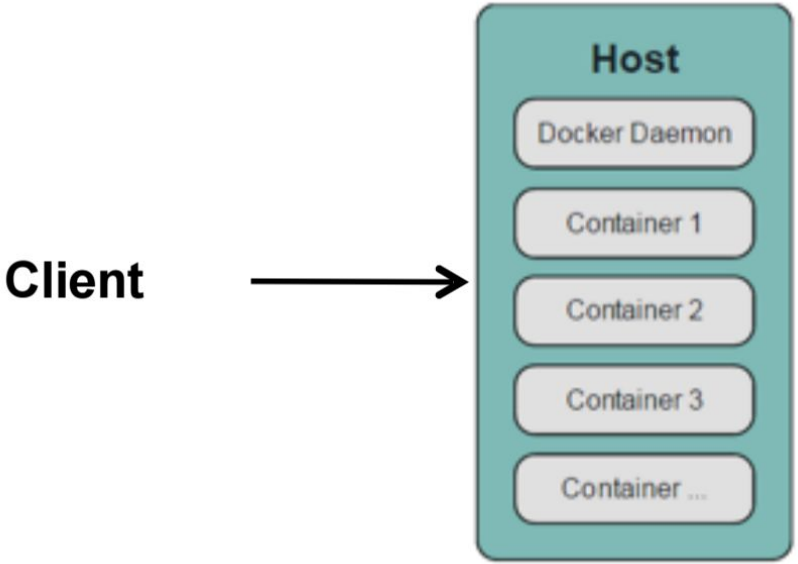
Docker and the Linux kernel

- **Docker Engine** is the program that enables containers to run
- Uses Linux kernel namespaces and control groups
- Namespaces limits what you can use
- Control groups limits how much you can use

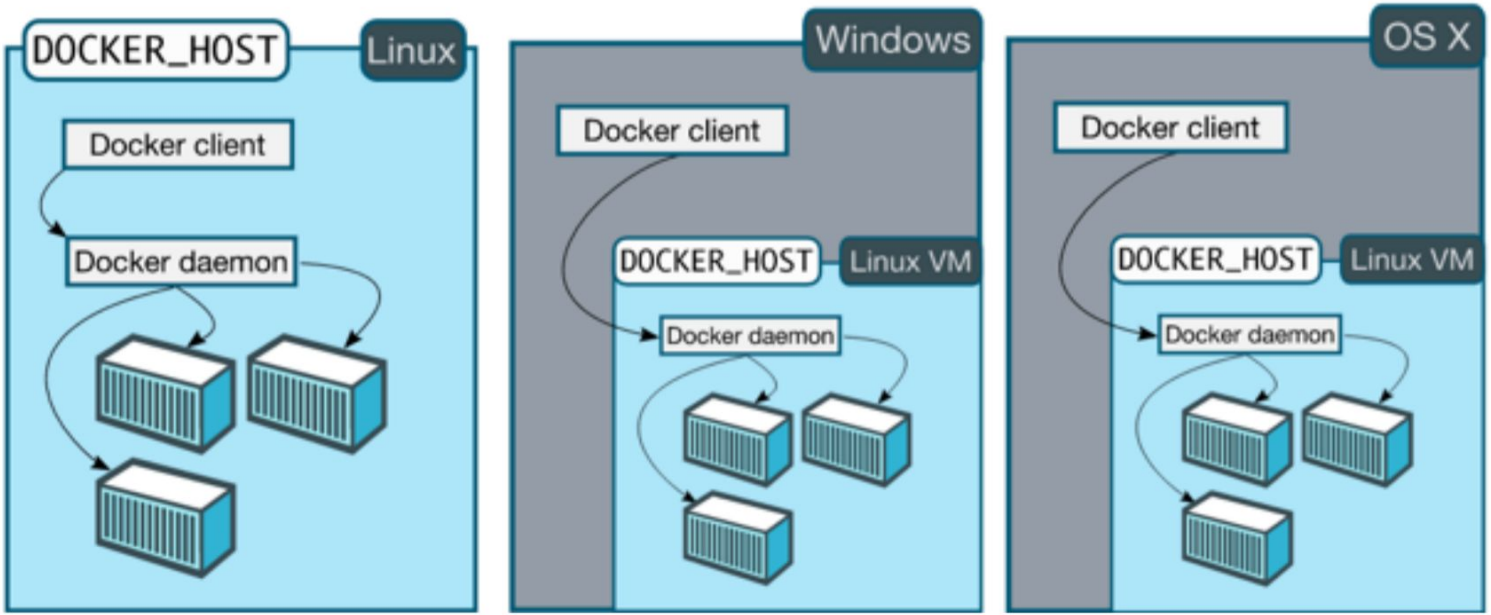


Docker client and daemon

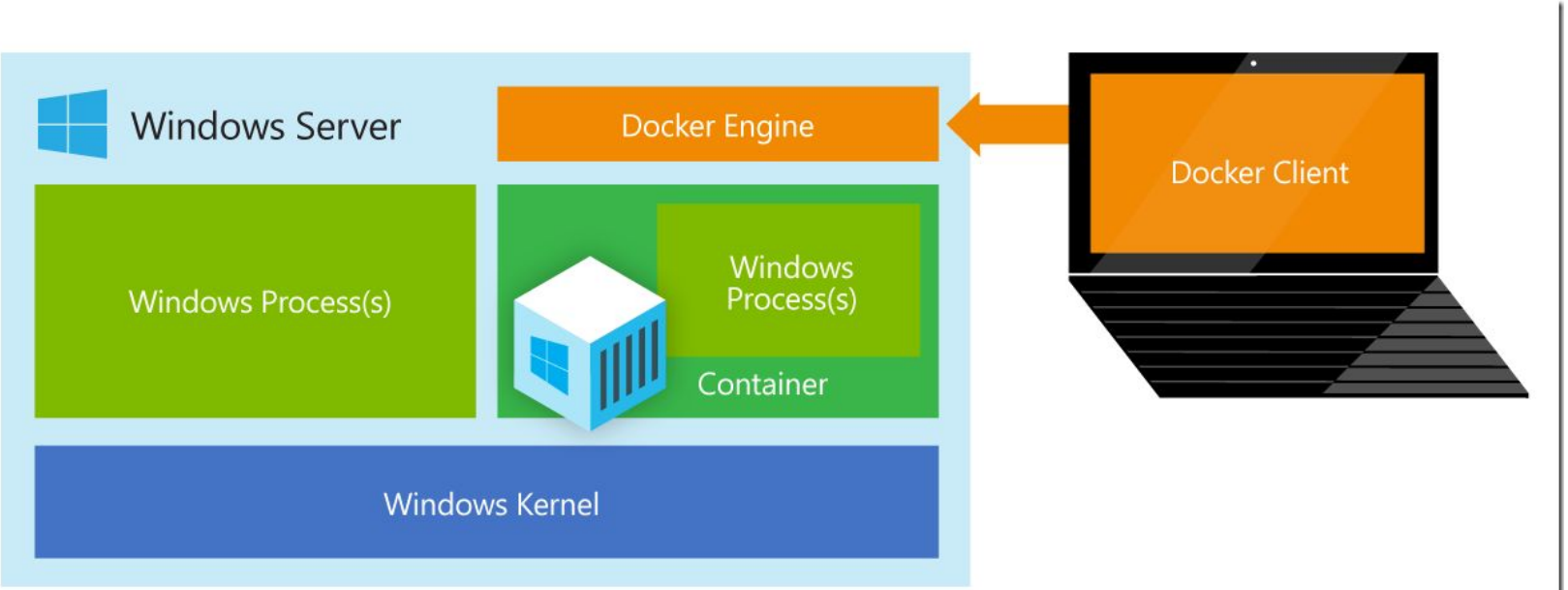
- Client sends user input to the daemon
- Daemon builds and runs containers
- Client and daemon on same host or on different hosts
- **Docker Machine** used to create hosts



Docker hosts



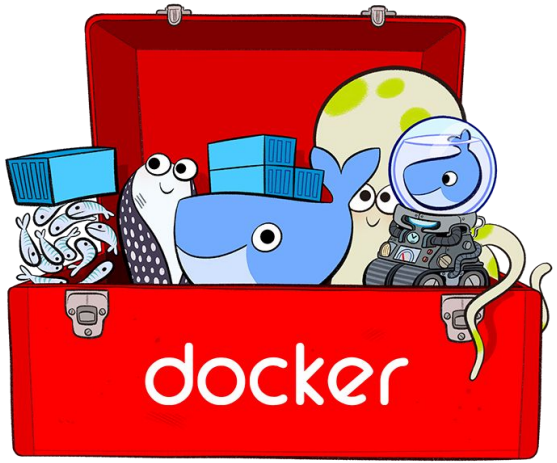
Docker on Windows Server 2016



PRAQMA

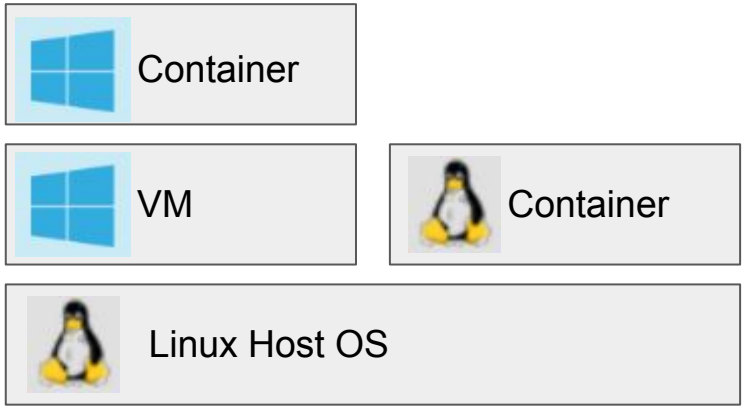
Installation Windows & Mac

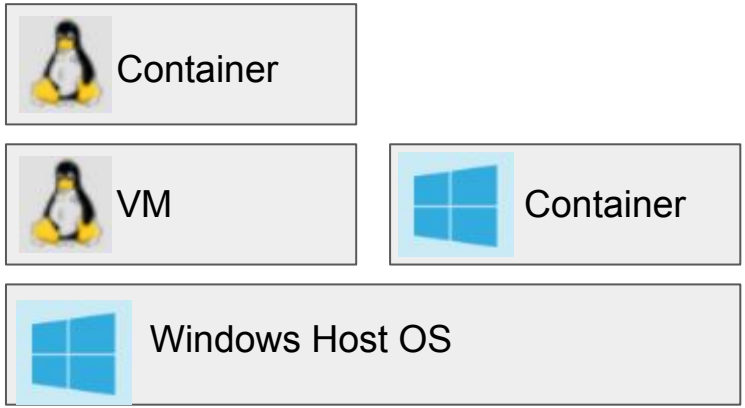
- Docker Toolbox
  - VirtualBox
- Docker Native Beta
  - Xhyve
  - Hyper-V
  - Alpine Linux



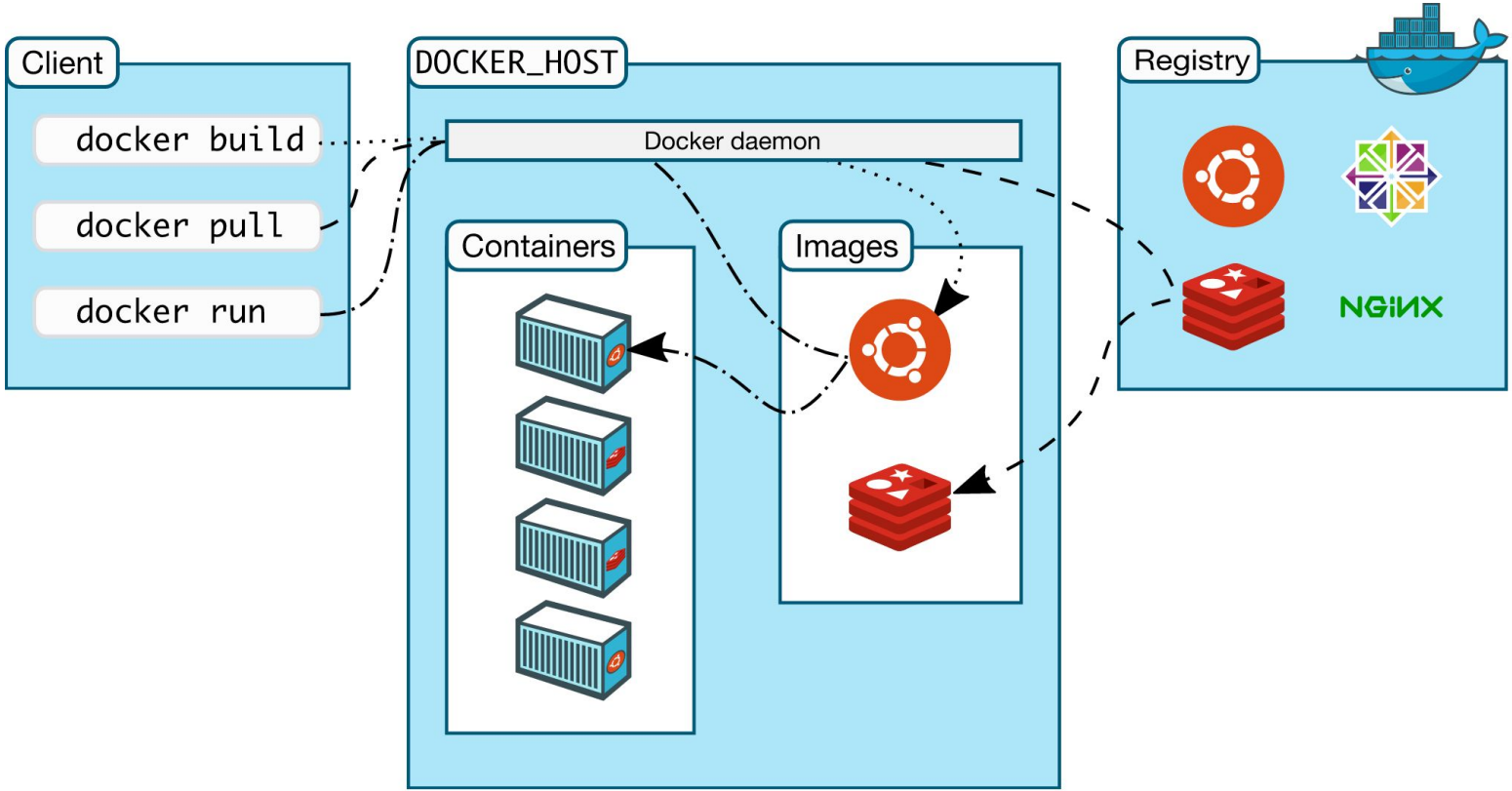
PRAQMA

Some host combinations



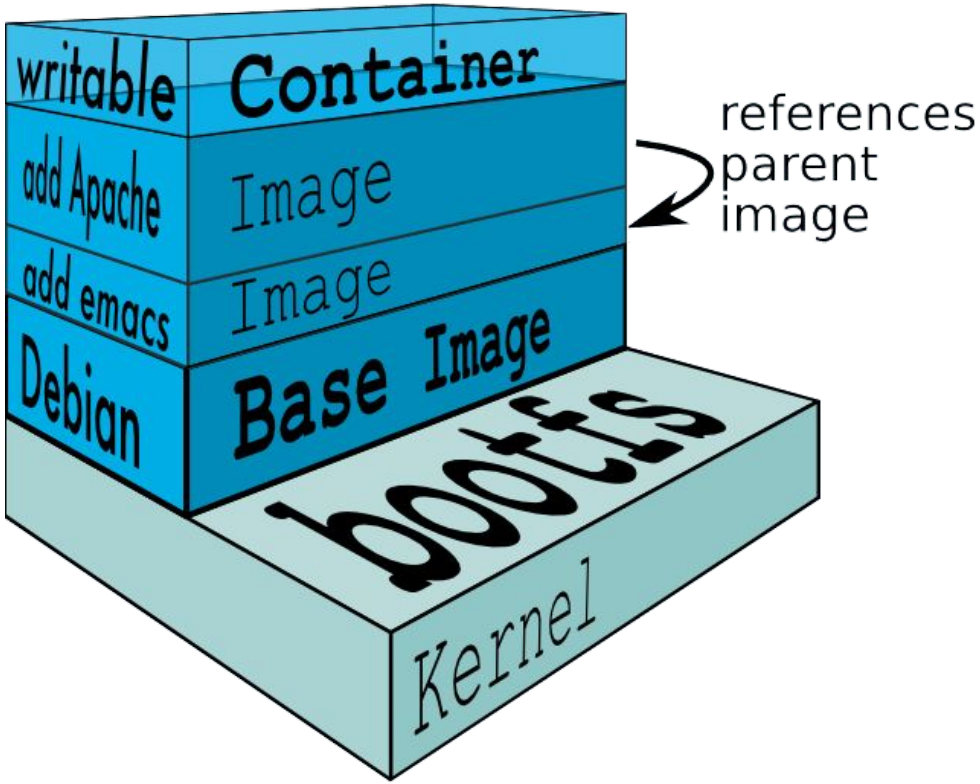


Architecture



PRAQMA

Layered images



Dockerfile

- Used to build images

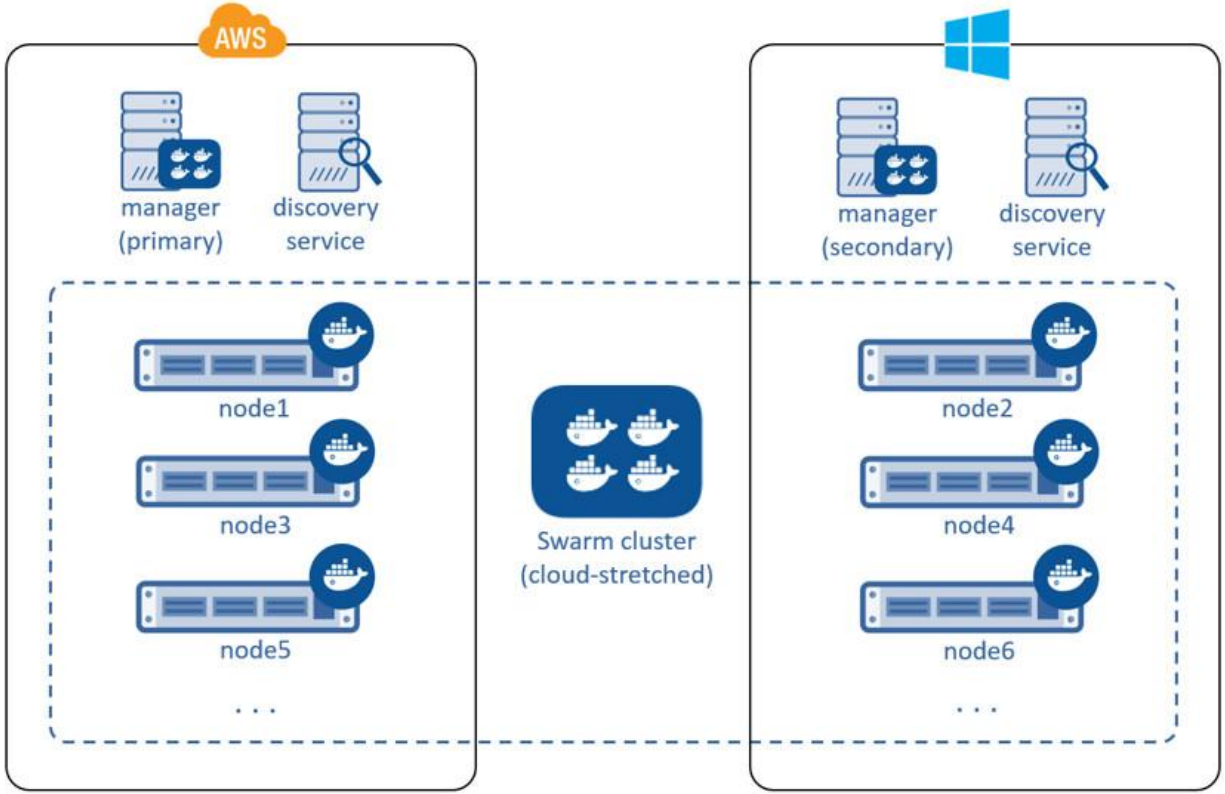
```
FROM alpine:3.3
MAINTAINER Praqma <info@praqma.com>

ENV http_proxy ${http_proxy:-}
RUN apk update && apk add libstdc++
COPY bootstrap.sh /bootstrap.sh
ENTRYPOINT ["/bootstrap.sh"]
```

Compose

- Multi-container applications
- YML configuration file

```yaml
version: '2'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Swarm


AWS
manager
(primary)
discovery
service
manager
(secondary)
discovery
service
node1
node3
node5
. . .
Swarm cluster
(cloud-stretched)
node2
node4
node6
. . .

Universal Control Plane

Docker Universal Control Plane

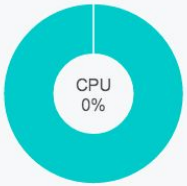
admin

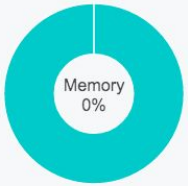
Dashboard

Overview

APPLICATIONS	CONTAINERS	IMAGES	NODES
0	9	8	1

Resources





Swarm Cluster

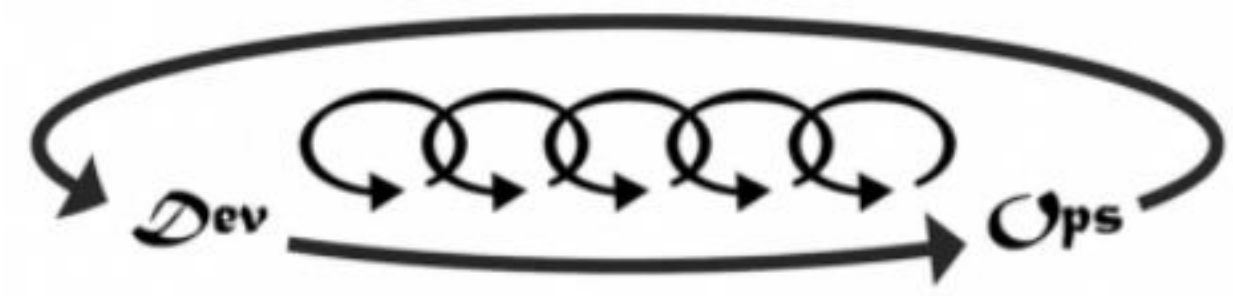
Status	Controller URL	Swarm Manager	Cluster Store
Healthy	https://10.0.2.15	tcp://10.0.2.15:2377	etcd://10.0.2.15:12379

Universal Control Plane 0.5.0 (fe1aaef) | API: 1.21

PRAQMA

Docker and the three ways of DevOps

1. Systems thinking: “The flow from left to right”
2. Amplify feedback loops
3. Culture of continual experimentation and learning



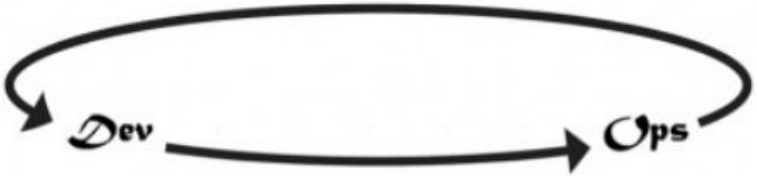
PRAQMA

The First Way: Systems Thinking

- Increase velocity
  - Docker images boot time
  - Convergence
  - Layered images
- Decrease variation
  - Throughout pipeline: Dev, integration, production
- Services isolated as containers provide better ownership
- Business outcome: Time to market

The Second Way: Amplify Feedback Loops

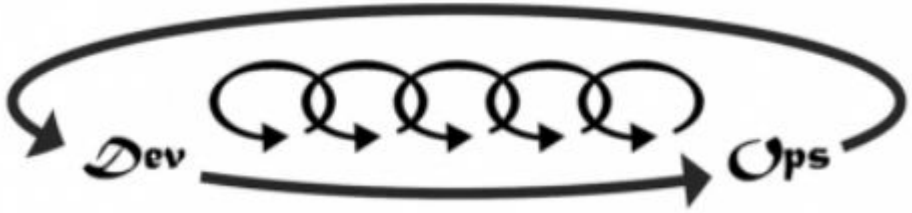
- A defect is not a defect unless it hits the customer
- Early discovery is less costly
- Complexity of infrastructure when defect is detected
- Image = immutable binary artifact
- Attach metadata:
  - When was it built, commit SHA, Git repo
  - How do I start, validate and monitor it
- Business outcome: Higher quality



PRAQMA

The Third Way: Continuous Learning

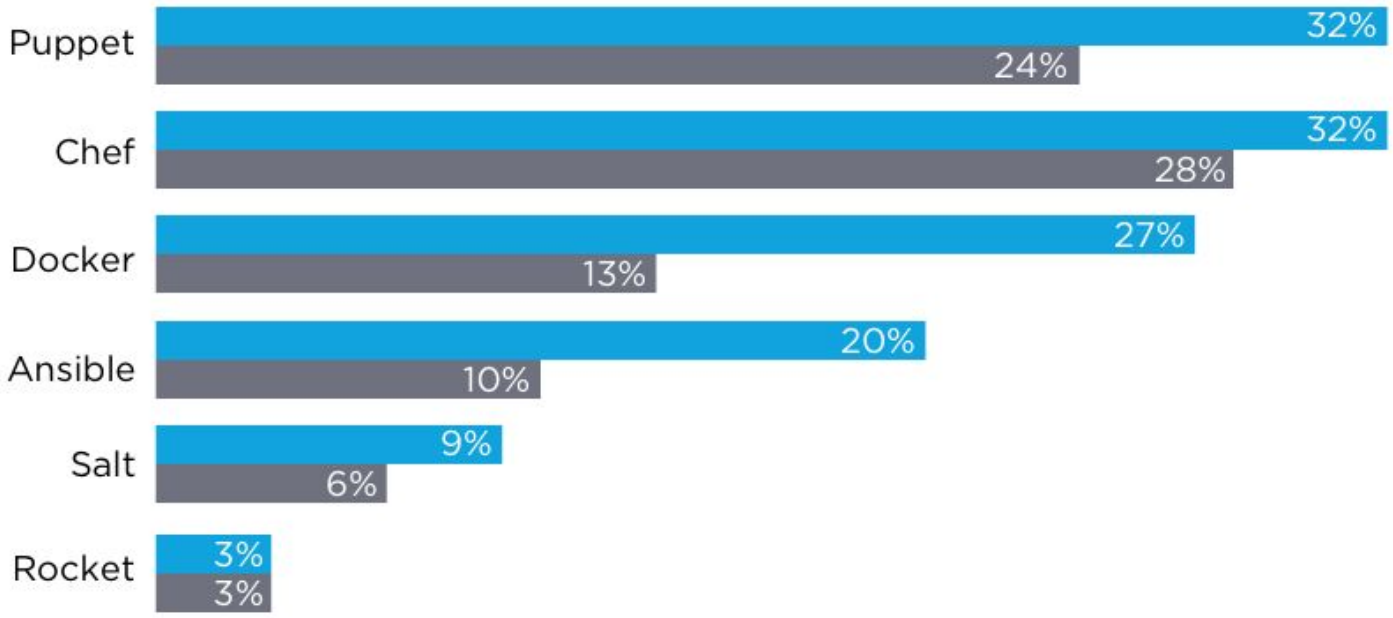
- Experiments and vision
- “Did the experiment produce results in the direction of the vision?”
- Setup “lab equipment” with prebuilt images
  - A Hadoop container ready to be fed data
  - Apache Spark container for other types of data
  - …
- Business outcome: Faster innovation



Trends

- RightScale “State of the cloud” survey, January 2016
- The Docker Survey, 2016

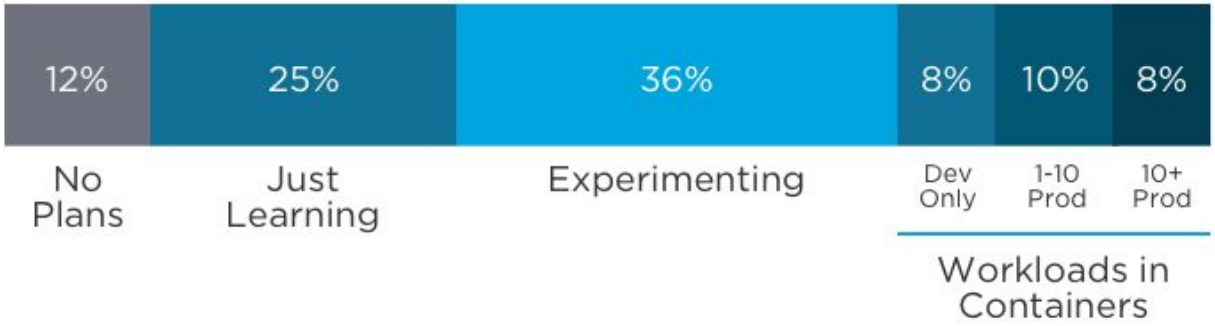
Respondents Using DevOps Tools



Source: RightScale 2016 State of the Cloud Report

PRAQMA

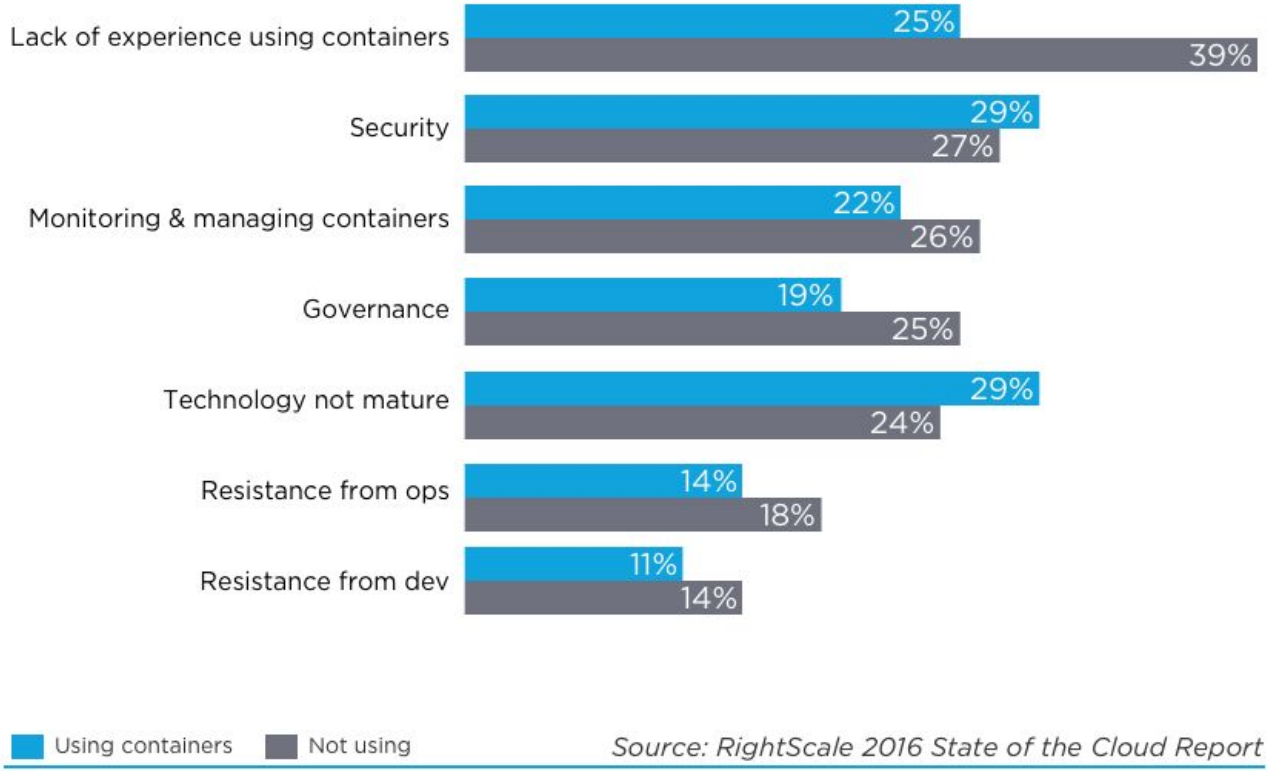
Container Usage of Respondents



Source: RightScale 2016 State of the Cloud Report

PRAQMA

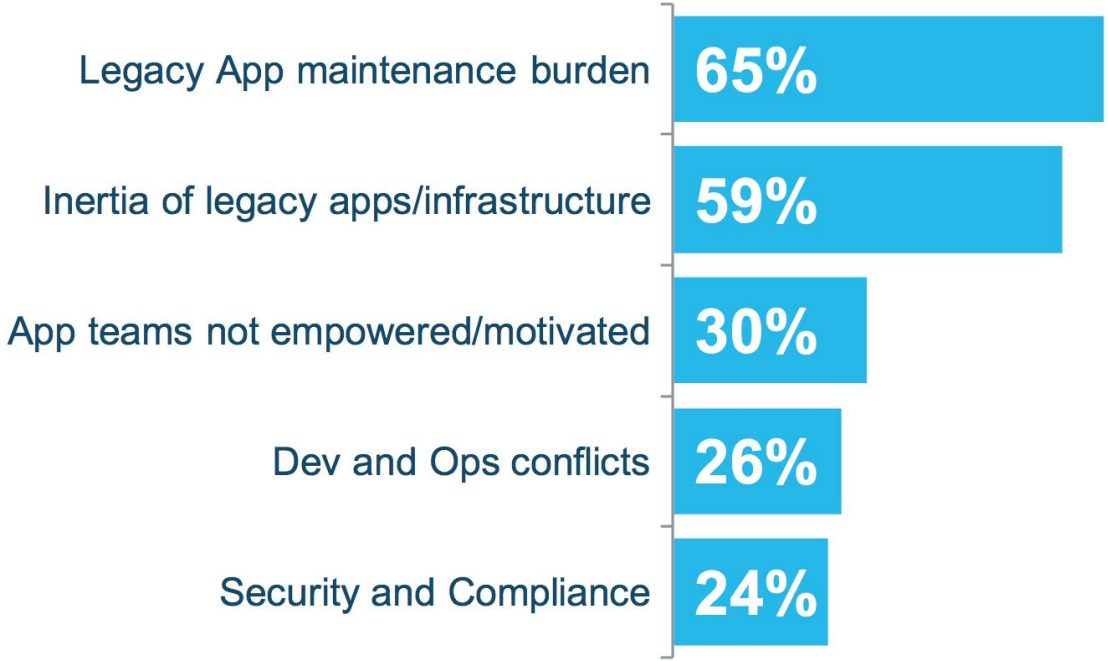
Container Challenges by Maturity



Source: RightScale 2016 State of the Cloud Report

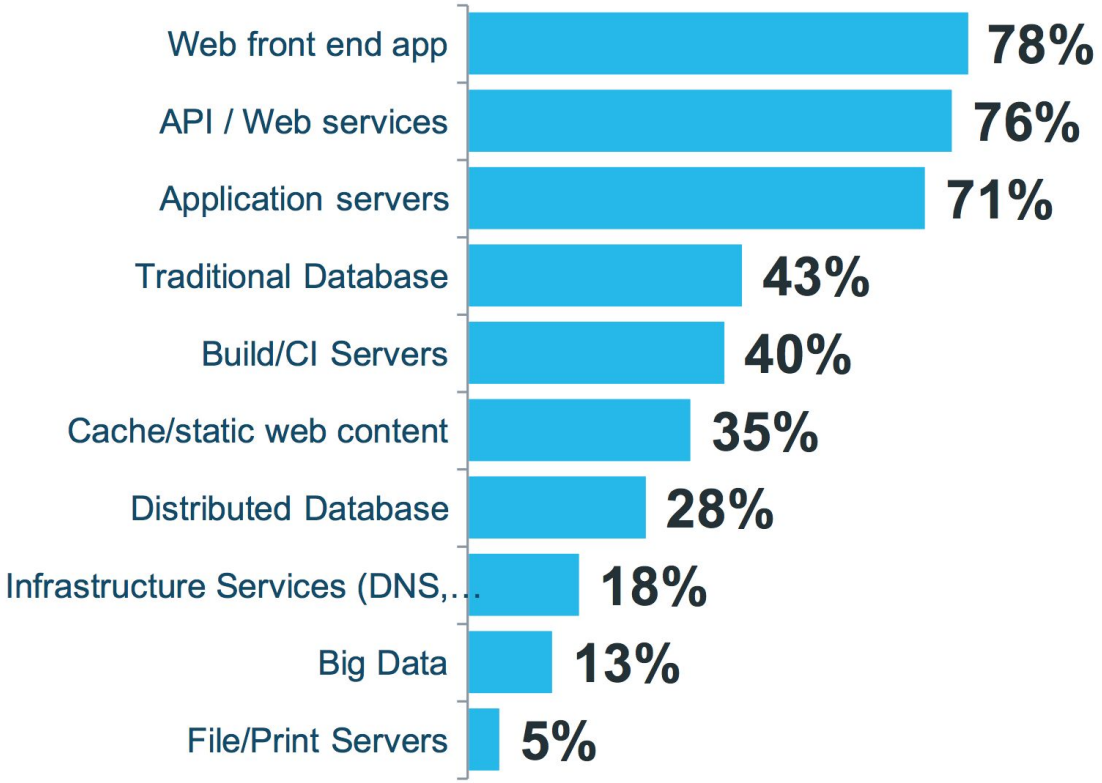
PRAQMA

Top Development Challenges



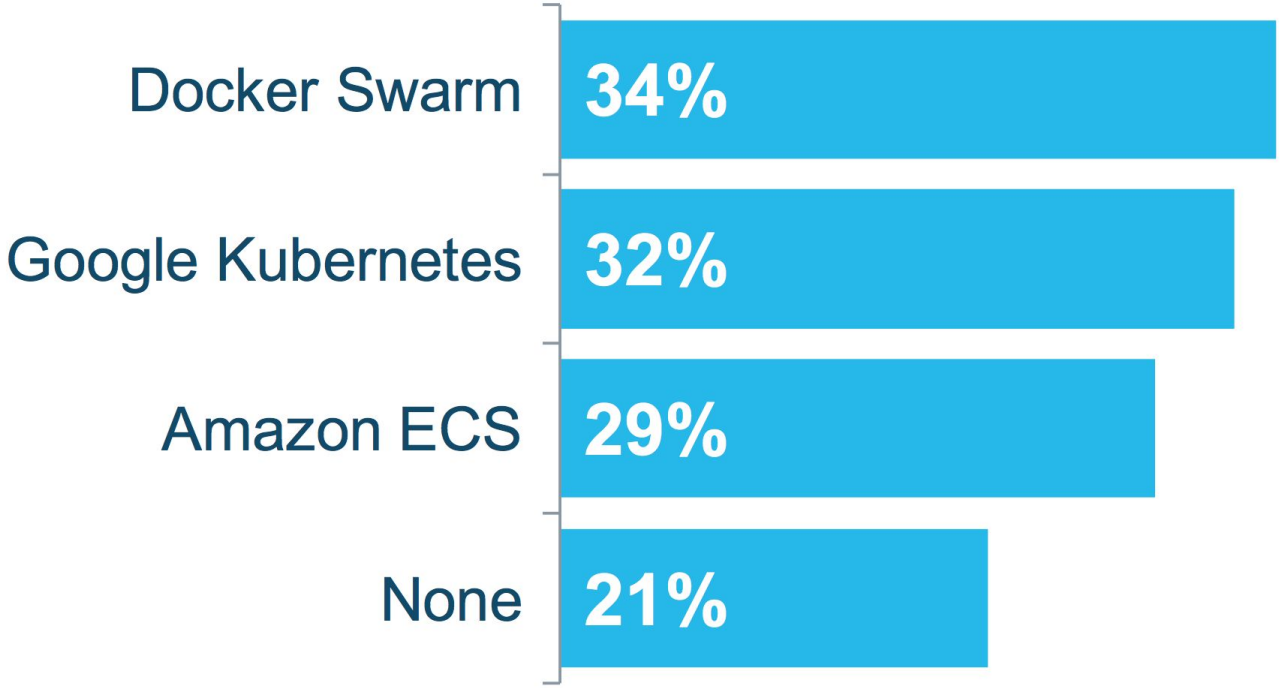
PRAQMA

Workload types in Docker



PRAQMA

What container orchestration and management solutions are you using/evaluating?



PRAQMA

Upcoming meetups

- Automation Nights Aarhus, 21. June
  - Implementing CD @ Systematic
  - Automation @ GoMore
- Docker Aarhus, 31. August